Hypercube Queuing Models in Emergency Service Systems: A State-of-the-Art Review

Maryam Ghobadi\(^a\), Jamal Arkat\(^b\), Reza Tavakkoli-Moghaddam\(^{c,d,*}\)

\(^a\) Ph.D. Candidate, Department of Industrial Engineering, University of Kurdistan, Sanandaj, Iran
\(^b\) Associate Professor, Department of Industrial Engineering, University of Kurdistan, Sanandaj, Iran
\(^c\) Professor, School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
\(^d\) LCFC, Arts et Métier Paris Tech, Metz, France

**Abstract.** This study provides a review of hypercube queuing models (HQMs) in emergency service systems (ESSs). This survey presents a comprehensive review and taxonomy of models, solutions and applications related to the HQM after Larson [12]. In addition, the structural aspects of HQMs are examined. Important contributions of the existing research are addressed by taking into account multiple factors. In particular, the integration of location decisions with HQMs for designing an ESS is discussed. Finally, a list of issues for future studies are presented.

**Keywords:** Hypercube queuing model; Facility location; Emergency service system.

1. Introduction

Emergency service systems (ESSs) provide the first care services when incidents occur and ensure public health and safety. In these systems, the customer's situation is usually critical and unstable. This means that delay in providing service may cause death or serious injuries. Given these conditions and, in general, the uncertainty in these problems, decision-making becomes more complex for managers. The design of ESSs requires strategic and tactical decisions [1]. The strategic decisions determine the number and location of servers. The dispatching policy that illustrates the decision about which server will respond to a request, and the server's coverage area are specified by tactical decisions.

ESSs can be classified into two main categories, namely customer-to-server and server-to-customer systems. In the first category, servers are immobile, and customers should visit them to receive a service. In the second case, servers are mobile and provide a service at the customer's location. As an example, in a case of fire, fire trucks are dispatched to the scene, and in emergency medical systems (EMSs), ambulances travel to the accident location. A system with mobile servers is called emergency response system. In a system with immobile servers, servers are usually considered indistinguishable from each other (e.g., seats of an airplane or beds of a hospital). In these cases, it does not matter exactly which server is busy, and only the number of busy servers is important [1]. On the other hand, mobile servers can be modeled as servers more precisely distinguishable from each other. That is, servers

\(^*\)Corresponding author. Tel.: +98 21 82084183; Fax: +98 21 88013102.
E-mail addresses: m.ghobadi@eng.uok.ac.ir (M. Ghobadi); j.arkat@uok.ac.ir (J. Arakat); tavakoli@ut.ac.ir , reza.tavakkolimoghaddam@ensam.eu (R. Tavakkoli-Moghaddam).
operate independently and can have different features (i.e., different preferences and mean service times), and a server's workload may be changed by the server's location. The hypercube queuing model (HQM), which was proposed by Larson [2] and used by many researchers, is a descriptive model suitable for modelling server-to-customer systems. Over the years, the model has been applied to a large number of public and private emergency systems, such as police departments and ambulance services. In this research, development history of the model and its basic ideas are reviewed as well as the implementation of this model is discussed.

1.1. Models description

The basic idea of this model is to develop the state space description of a queuing system to use complex dispatching policies and illustrate each server individually. These models consider the spatial and temporal complexity of the area under study and are suitable for centralized systems. In a centralized system, each customer calls the central unit (i.e., dispatcher), and it dispatches the first idle server to that customer, according to a preference list. This list is prepared based on factors, such as distance and customer requirements. For example, if the list is ordered based on the distances between customers and servers, then once an emergency call is received, the closest server is dispatched. If the closest server is busy, then the second, third or closest available server is dispatched instantly. Therefore, in the case of mobile servers, system workload is shared between servers better than in cases with immobile servers. Also, if there are no available servers, the customer enters a waiting line or is transferred to another ESS.

The term hypercube is taken from the space that describes the states of the servers. At any point of time, each server is free (i.e., 0) or busy (i.e., 1). Therefore, there are $2^N$ states in a system with $N$ servers. A certain state of the system is specified by a list of free and busy servers (an array of 0s and 1s). For example, the state \{011\} corresponds to a 3-servers system, in which server 1 is free and servers 2 and 3 are busy (reading from left to right). For $N = 3$, the state space can be shown by a cube (Fig. 1), in which each vertex indicates one state of the system. For $N > 3$, the state space becomes a hypercube.

{Please insert Fig. 1 about here.}

In an HQM, the performance measures of the system are obtained by calculating limiting probabilities, such as calls per hour, mean travel time, mean response time, mean workload, maximum workload imbalance, fraction of customers answered by primary servers, and fraction of customers answered by backup servers. To determine the limiting probabilities, $2^N$ balance equations should be solved. This is simply done by using the flow-balance criterion around the states of the system. Accordingly, in a steady state, the rate at which the system enters state $i$ is equal to the rate at which the system leaves that state. By entering or exiting a state, a transition occurs. Actually, a transition arises when a server's state changes from busy to free or contrariwise. Each transition takes place probabilistically over an edge of the hypercube. When a customer is served by a server, a downward transition happens. Thus, the rate of downward transition is equal to the service rate. An upward transition occurs once
a free server is selected to be dispatched to a customer for service. The rate of an upward transition is determined by a set of server assignment and dispatching policies [2].

To better understand the presented concepts, an example is provided here. Consider a simple network with three atoms connected by a one-way street (Fig. 2). The distance matrix between these atoms is presented in Table 1.

{Please insert Fig. 2 about here.}

{Please insert Table 1 about here.}

It is assumed that the center of each atom is the location of a server, and a fixed-preference dispatch policy is in use. That is, when a call is received, a dispatcher assigns the first available server from a dispatch list ordered from the most preferred to the least preferred for that call. The preference matrix based on the shortest travel distance is shown in Table 2.

{Please insert Table 2 about here.}

To define the hypercube state probabilities, $2^3$ balance equations are written. If $\lambda_i$ represents the arrival rate of customers from atom $i=(1, 2, 3)$, and $\mu_j$ indicates the service rate of server $j=(1, 2, 3)$, then $\lambda=\lambda_1+\lambda_2+\lambda_3$ and $\mu=\mu_1+\mu_2+\mu_3$, where $\lambda$ and $\mu$ are the total arrival and total service rates, respectively.

The following is an explanation of how to build the balance equation for a certain state, like $\{101\}$. The system leaves state $\{101\}$ if a customer arrives or server 1 or 3 completes its service, so the transition rate is $(\lambda+\mu_1+\mu_3) P\{101\}$, where $P\{101\}$ shows the probability that the system is in state $\{101\}$. Moreover, the system enters this state in one of the following three ways: i) from state $\{001\}$ if a customer arrives from atom 1 or 3 (in accordance with Table 2), ii) from state $\{100\}$ if a customer arrives from atom 3 and iii) from state $\{111\}$ when the service of server 2 is completed. The transition rate is $(\lambda_i+\lambda_3) P\{001\}+\lambda_i P\{100\}+\mu_3 P\{111\}$. Since the transition rates of the system out of and into a state are equal in a steady state, the balance equation of state $\{101\}$ is written by:

$$(\lambda+\mu_i+\mu_3) P\{101\}=(\lambda_i+\lambda_3) P\{001\}+\lambda_i P\{100\}+\mu_3 P\{111\}$$

(1)

The balance equations for other states can be written in a similar manner. To find out more, see Chiyoshi et al. [3], who presented a set of hypercube models with different assumptions and indicated steady-state equations and some practical specifications of each model.

In the next section, a brief introduction of the exact and approximate HQM is provided. Section 3 classifies the existing papers in the literature according to their assumptions and highlights the contributions of the model formulations and solution approaches. Section 4 provides the details of studies, which are incorporated the HQM in the location problem. Finally, this paper is ended by discussing about future research directions. There are also some papers that cannot be classified in the following categories; however, they are very helpful and prepare basic concepts for the HQM. Potential applications of the HQM, how it
works, when it is preferred to other models and the required resource are given in Chaiken [4]. Larson [5] presented a manual for users of the model. Larson [6] prepared a list of computer programs and provided information for users. Sacks [7] and Sacks [8] developed a software, named desktop hypercube, and evaluated its performance in a case study. Larson [9] examined the performance of operations research in homeland security and introduced the HQM as an efficient tool in this area. Galvao and Morabito [1] reviewed probabilistic models for the design of emergency service systems. They also surveyed the extensions of these models, which embedded into the HQM.

2. Exact and approximate HQMs

As the first HQM, Larson [2] analyzed a multi-server queuing system with distinguishable servers, which they support each other. He also developed a computationally efficient algorithm to evaluate the model analytically and calculate several performance measures. His model was designed for location and districting problems in urban emergency systems; districting is defined as partitioning an area into sub-areas (i.e., districts) according to its features. The following assumptions are considered in the original HQM [10, 11].

1) The district of a server is an area that is handled by the server, of course if it is available. Otherwise, customers in that area will be responded by a server out of the district. If all servers are busy, then the customer enters a queue or is served by another ESS. Also, there may be more than one server in each district that share the workload of that area.

2) Each service area is divided into sub-areas, called atoms. This classification can be based on the census report, urban areas and so on. Demand points are located at the center of each atom.

3) Demands of each atom are generated independently via a Poisson process with known rate $\lambda_i$.

4) There is at least one server in each atom, and exactly one server is dispatched to serve a customer.

5) Server assignment takes place according to a fixed-preference procedure. For each atom, there is an ordered list of preferred servers to dispatch. The dispatcher searches that list in order and sends the first available server. This list is usually obtained by geographical measures (e.g., travel times); however other criteria (e.g., allocating expert personnel) can be considered.

6) Service time follows an exponential distribution with a known rate. Brandeau and Larson [12] showed that service time generally includes travel time, on-scene time and maybe some follow-up time (Fig. 3).

{Please insert Fig. 3 about here.}

For the convenience of the reader, a summary definitions of symbols which frequently used in this paper is presented in Table 3.

{Please insert Table. 3 about here.}
2.1. Approximate HQM

Each system is evaluated by its performance measures. Larson [2] divided these measures according to the characteristic that they are evaluated. As an example, the mean region-wide travel time (considering all types of calls), workload imbalance, and fractions of dispatches that are inter-district dispatches are suitable performance measures from a region-wide perspective; workload (fraction of time that server is busy), mean travel time and fraction of responses of each response unit that are inter-district can measure the performance of a response unit (server). It is possible to focus simultaneously on several performance measures as objectives while the other measures maintained at an admissible level.

There are three main ways to evaluate the performance of a system [13]: 1) exact approaches (e.g., HQM proposed by Larson [12]), 2) discrete-event simulation, and 3) approximate approaches (e.g., approximate hypercube (AH) model proposed by Larson [14]). The advantages of the approximate procedures in comparison with two other ones are that their computation time is low and is not influenced by the features of the system.

The AH model is a simple iterative procedure to estimate the performance measures of a system. In a system with \( N \) servers, the approximate model requires only \( N \) equations rather than \( 2^N \), as it is necessary in the original model. Although, in practice, it has been approved that the solution of the AH model is good enough, it uses simplified assumptions such as no cooperation between servers. Furthermore, in most approximate approaches presented so far, it is assumed that only one server is located in each base. Regarding these two assumptions, the state of each server (i.e., free or busy) is independent of the state of other servers; therefore, assuming non-cooperation can be considered almost equivalent to assuming independent servers. Hence, the HQM is reduced to an \( M/M/N \) queuing system. In this system, if state \( S_k \) represents that \( k \) servers are busy exactly, and \( P\{S_k\} \) is the probability that the system is in state \( S_k \), the steady-state probabilities are as follows:

\[
P\{S_k\} = P_k = \frac{N^k \rho^k P_0}{k!}, \quad k = 1, 2, \ldots, N - 1
\]

\[
P\{S_N\} = P_N = \frac{N^N \rho^NP_0}{N!(1 - \rho)}
\]

\[
P\{S_0\} = P_0 = \frac{1}{\sum_{i=0}^{N-1} \frac{N^i \rho^i}{i!} + \sum_{i=0}^{N} \frac{N^N \rho^N}{i!}}
\]

where \( \rho = \frac{\lambda}{N\mu} < 1 \) , if \( \lambda \) and \( \mu \) indicate the demand and service rates, respectively. These results can be extended for the \( M/M/N/\infty \) queuing system.

In the AH model, the probability that the \( i \)-th customer is served by the \( j \)-th server in his/her preference list is equal to the probability that the first the \( j-1 \) servers are busy and the \( j \)-th server is the first free server. In Larson [14], servers are selected randomly until an idle server is found. If \( B_j \) indicates the event that the \( j \)-th selected server is busy and \( F_j = B_j^c \)
indicates the event that the $j$-th selected server is free, then $P\{B_1B_2\ldots B_jF_{j+1}\}$ shows the probability that the $(j+1)$th selected server is the first idle server. Selection of servers is done in a completely random process without replacement. Therefore, each server probability of being busy is $\rho$, and the probability that the $(j+1)$th selected server is the first idle server is $\rho^j(1-\rho)$, if servers are independent. Thus, Larson [14] presented a factor, $Q(N,\rho,j)$, to correct the results for the case, in which the servers are dependent (or, in other words, there is cooperation between the servers).

$$P\{B_1B_2\ldots B_jF_{j+1}\} = P\{F_{j+1} | B_1B_2\ldots B_jS_k\} P\{B_j | B_1B_2\ldots B_{j-1}S_k\} \ldots P\{B_1 | S_k\}$$

$$= Q(N,\rho,j)\rho^j(1-\rho)$$

$$Q(N,\rho,j) = \frac{P\{F_{j+1} | B_1B_2\ldots B_j\} P\{B_j | B_1B_2\ldots B_{j-1}\}}{(1-\rho)} \frac{P\{B_1 | S_k\}}{\rho} \ldots \frac{P\{B_1 | S_k\}}{\rho}$$

$$= \frac{\sum_{k=j}^{N-1} (N-j-1)!(N-k)\binom{N}{k}^j \rho^{k-j}}{(1-\rho)N! \rho^j} \cdot \frac{N^i}{\rho^j}$$

Jarvis [15] extended the approximation procedure in [14] and presented an algorithm for loss systems (zero-line capacity). In his procedure, service time distribution is depended on the type of server and customer. Different types of customers have different demand rates, and service rate is different for each pair of server and customer. He also estimated server workload more accurately than Larson [14]. Furthermore, he could show that although the shape of service time distribution was not completely ineffective on the results of the HQM, its influence was very small [15].

The details of the Jarvis algorithm, which approximates the busy probabilities of each server, are presented below [16].

In the initial step of the algorithm, $\rho_j$ is calculated by assuming that every customer is assigned to its first preferred server. Therefore, there is no cooperation between servers.

$$\rho_j = \sum_{c \in \alpha_k} \lambda_c \tau_{jc}$$

$$\tau = \sum_{c=1}^{C} \frac{\lambda_c}{\Lambda} \tau_{jc}$$

**Step 1:** Calculate the correction factor using Eq. (4);

**Step 2:** Calculate an approximation for server workload for $j=1,\ldots,N$;
\[
\rho_j(\text{new}) = \frac{V_i}{V_i + 1}
\]
\[
V_i = \sum_{k=1}^{N} \sum_{c,d,a=i}^{j} \lambda_c \tau_{kc} Q(N, \rho, k - 1) \prod_{i=1}^{k-1} \rho_{ai}
\]

**Step 3:** Stop if maximum change in \( \rho \) is less than convergence criterion;

**Step 4:** Calculate the following equation;

\[
P_n = 1 - \frac{\sum_{j=1}^{N} \rho_j}{N \rho}
\]
\[
f_{j_c} = Q(N, \rho, k - 1)(1 - \rho_j) \prod_{i=1}^{k-1} \rho_{ai}
\]
\[
\tau = \sum_{c=1}^{C} \frac{\lambda_c}{\lambda} \sum_{j=1}^{N} \tau_{jc} f_{j_c} \frac{1 - P_N}{1 - P_N}
\]

**Step 5:** Return to Step 1.

### 3. Classification of the HQM

The HQM can be classified from different perspectives. From a general point of view, HQMs are categorized based on the dispatching policy, backup strategy and server type. Dispatching strategies are divided into two general categories, namely single and multiple. In single dispatch strategy, it is assumed that only one server is needed to respond to the demand of a customer. In multiple dispatching strategy, customers require two or more servers simultaneously. As an example, in a sever car accident, several low-level ambulances are sent for transportation, along with one or two high-level ambulances to provide more professional medical services, and in the case of police patrol, two police cars with one or two officers in each are usually sent.

Backup strategies are divided into two general categories, namely total and partial. In a total backup strategy, a customer is lost or enters the queue only if there are no idle servers. That is, as long as there is an idle server, customers will not be queued. In a partial backup strategy, only some servers can respond to each customer. For instance, a server may be able only to serve customers who are located at a certain distance from that server. Therefore, in these models on the arrival of a customer, if its backup servers are busy, the customer is lost or enters a queue, even if there is a free server.

In this survey, the term homogeneous stands for servers with the same rate. A service rate depends on many factors. For example, for trained servers, the service rate is usually more than newcomers, or, in some cases, the location of customers and servers influences the travel time, which is a part of service time. Also, a service rate may be different for various types of activities; as an example, in police patrol systems, two kinds of activities are defined, namely dispatching to call for service (CFS) and patrol initiated activity (PIA). Officers in police
patrol cars control their patrol areas to improve public safety, and they may check buildings, cars or people; these actions are called PIA. Such activities can also be defined in medical systems. Suppose that a patient visits an ambulance base for receiving service. Although, in this case, the server has not been sent by a dispatcher, it would be busy and spend considerable time. Therefore, these activities have different service rates and must be taken into account in calculating the performance measures.

Based on the aforementioned explanations, studies in the field of HQMs can be divided into eight categories. The point that should be noted, is that, in cases where servers are not supposed to support each other, they should be treated as non-homogeneous servers and hence we just review six distinguishable categories below.

3.1. Single dispatch, total backup and homogeneous servers

The assumptions considered in this part are similar to those of the original model. After Larson [2], Larson and Franck [10] evaluated the performance measures of an emergency response system where the dispatcher accesses automatic vehicle location (AVL) systems. AVL systems estimate the current location of servers in the service area and help the dispatcher to forward the closest available server to each customer. Unlike in the studies with a fixed-preference dispatching strategy, in this study, the upward transition rate depends on the real time locations of idle servers. Therefore, to determine the matrix of upward transition rates, a recursive method is used. According to this method, geographical locations of customers are first fixed, and then the hypercube vertices are visited in a unit-step procedure. This matrix is completed when all vertices of the hypercube are visited once for each geographical area.

Chelst and Jarvis [17] proposed an extension of the HQM where the probability distribution of travel time is calculated, in addition to its average. Larson and Rich [18] investigated the relationship between travel times and dispatching policy in a police department. They found that travel times would not increase remarkably as the service area of each server increased.

Souza et al. [19] proposed a modified HQM in the context of emergency systems and considered priority for the customers in the queue, based on the degree of severity. In their work, high-level customers are those whose lives are in risk and need more advance equipment and more specialized medical team in comparing with the low level customers. They assumed a non-preemptive priority discipline when a server becomes idle, it serves a low-level customer in the queue if there are no high-level customers. In this study, a layering procedure [20, 21] is used to take into account different classes of customers. In this procedure, the total service area is divided into different sub-areas, called atoms, and each atom is distributed into sub-atoms each of which is an independent source of customers from one of the priority classes. There is an example of such systems with infinite queue capacity. Consider a queue system with three priority classes, $a$, $b$ and $c$ where $a$ and $c$ represent the highest and lowest priority classes, respectively. The dispatching matrix is shown in Table 4, in which each atom has three layers. If there are two customers in the queue, then all possible states of the queue are: $\{aa\}, \{ab\}, \{ac\}, \{bb\}, \{bc\}$ and $\{cc\}$. If each class $r$ ($r = a$, $b$ and $c$) of customers arrives according to the Poisson process with rate $\lambda_r$, and
service time is distributed exponentially with rate \( \mu \), then the transitions into and out of the queue state \( \{ab\} \) are as shown in Fig. 4. It is obvious that according to the priority queuing discipline, when a server becomes free, the only transition is \( \{ab\} \rightarrow \{b\} \), and transition \( \{ab\} \rightarrow \{a\} \) is not allowed. The balance equation around the state \( \{ab\} \) is built as usual by setting the transition rate into a state equal to the transition rate out of that state.

{Please insert Table 4 about here.}
{Please insert Fig. 4 about here.}

In addition to the studies presented above, some studies have integrated the HQM and location models [22, 23]. The performance of emergency systems is associated with the location of servers and their allocation to the customers. Therefore, optimizing these two problems can improve some performance measures (e.g., mean travel time) simultaneously.

Daskin [24] formulated the maximal expected coverage location problem (MEXCLP), where servers are located optimally to maximize the expected coverage of demand in a situation, in which some servers may be unavailable. He recognized the busy probability of each server \( \rho \), which can be calculated by an Erlang loss system equation, \( \rho = \lambda / N \mu \). Therefore, if each demand point is covered by \( n \) servers, then the probability that a demand is covered by at least one server is equal to \( 1 - \rho^n \). The formulation of this problem is as follows. In this model, some simplifying assumptions (e.g., independent servers and the same busy probability for each server) are considered.

\[
\max \sum_{j=1}^{M} \sum_{i=1}^{N} (1 - \rho) \rho^{i-1} h_i y_{ij} \tag{12}
\]

s.t.
\[
\sum_{j=1}^{N} y_{ij} - \sum_{j=1}^{M} a_{ij} x_j \leq 0, \forall i \tag{13}
\]
\[
\sum_{j=1}^{N} x_j \leq N \tag{14}
\]
\[
x_j = 0, 1, \ldots, N, \forall j; \quad y_{ij} = 0, 1, \forall i, j \tag{15}
\]

where \( N \) is the maximum number of facilities, \( M \) is the number of demand points, and \( x_j \) is the number of servers in facility \( j \). Also, we have:

\[
y_{ij} = \begin{cases} 1 & \text{if node } i \text{ is covered by at least } j \text{ facilities} \\ 0 & \text{otherwise} \end{cases}
\]

As mentioned before, the objective function (12) maximizes the expected number of demands that are covered. Constraint (13) calculates how many times demand point \( j \) is covered. Constraint (14) limits the maximum number of facilities that can be deployed, and Constraint (15) shows that more than one server can be assigned to each facility.
Batta et al. [25] proposed an adjusted MEXCLP (AMEXCLP) by relaxing three basic assumptions of the MEXCLP, independent servers, the same busy probability for servers and independence between busy probability of servers and their locations. The formulations of the AMEXCLP and MEXCLP are similar, except for the objective function. The objective function of the AMEXCLP, which relaxes the independence assumption using the Larson’s correction factor, is as follows:

\[
\text{Max} \sum_{j=1}^{N} \sum_{i=1}^{M} Q(N, \rho, j-1)(1-\rho)\rho^{i-1}h_{ij}
\]

where \( y_{ij} \) is one if node \( i \) is covered by at least \( j \) servers, and otherwise it is zero, and \( Q(N, \rho, j-1) \) is computed by Eq. (4). They also integrated the HQM with a heuristic optimization procedure to find a set of locations for servers, which maximizes the expected coverage. They concluded that there is a conflict between the results of the AMEXCLP and those of a hypercube optimization procedure. Furthermore, Chiyoshi et al. [26] showed that the MEXCLP and its aggregate version (i.e., AMEXCLP) with the HQM are not comparable, because the structures of their objective functions are different, and the MEXCLP cannot consider queued customers. Furthermore, for queued customers, the waiting time plus travel time may exceed the critical covering time. In an HQM with infinite line capacity, these customers are served and added to the system’s workload; however, in the MEXCLP and AMEXCLP, they are not even covered. They investigated these two points in the study carried out by Batta et al. [25]. Galvao et al. [27] compared the MEXCLP and the maximum availability location problem (MALP). The MALP is proposed by Revelle and Hogan [28] maximizing the number of customers that can be covered in a target response time with reliability of \( \alpha \). Both the MEXCLP and MALP are probabilistic extensions of the maximum covering location problem (MCLP), and actually, they are two different perspectives of the same concept. They also proposed an extension of the MALP (EMALP), in which each server has a different busy probability. Furthermore, in the EMALP, the Larson’s correction factor is used to account for dependency between servers. Finally, these two extended models are solved by simulated annealing (SA). Chiyoshi et al. [26] developed Tabu Search (TS) (Glover and Laguna, [29]) for the EMALP and compared the results of this algorithm against those of the SA algorithm developed by Galvao et al. [27]. They showed that the solutions of SA outperforms TS for small networks in terms of quality, while TS performs better than SA for the larger networks generated randomly. The hypercube queuing model is used to calculate the server’s busy fractions.

Goldberg et al. [30] developed a model to locate emergency facilities. The goal of their model is to maximize the expected number of customers who are responded to within eight minutes (success rate). They used the Jarvis’ procedure to estimate server utilization rate. They also proposed a model to estimate travel time distribution between each pair of server and customer in a case study.

McLay and Mayorga [31] proposed a location model to maximize two performance measures: 1) the expected number of customers who are survived and 2) the expected number of customers who are responded to within the specified time threshold. These two performance measures are functions of response time, which is affected by the distance between servers.
and customers. These measures are evaluated only for customers whose lives are threatened, and the Larson's approximation algorithm is used to estimate them. They showed that patient survival rate optimization is related to how the response time threshold (RTT) is chosen. Toro-Diaz et al. [32] proposed a non-linear mixed-integer optimization model to find the location of ambulances (see Section 3). They found that the use of closest dispatching policy enables the model to minimize the response time and maximize the coverage.

Usually, in an EMS, servers are located in a fixed base. As population grows, demands for ambulances have increased, and more ambulance bases are required. Demands for ambulances are time dependent and may be changed weekly, daily and even hourly; thus building permanent bases to cover variable demands is costly. In response to demand fluctuation, redeployment strategies are used to change the location of ambulances dynamically [33, 34]. There are two types of redeployment strategies: 1) multi-period: in this strategy, the volumes of demands are predicted for different sectors of the service area and for different periods of time; then, ambulances are redeployed to face with demand fluctuation, and 2) real-time: in this strategy, when one or more ambulances are dispatched, the other available ambulances are redeployed to guarantee the desired coverage. Sudtachat et al [35] proposed a dynamic relocation strategy by using a nested compliance table. This table indicates where ambulances should be located when there are a certain number of ambulances available. Actually, ambulance stations are specified as a function of the state of the system by a compliance table. A nested compliance table restricts the number of relocations that can occur simultaneously. They extended an integer programming model to determine an optimal nested compliance table strategy and maximize the expected coverage. The relocation and approximation of steady state probabilities are input parameters of this model. Finally, they compared this dynamic strategy with a non-relocation model (AMEXCLP), which is proposed by Batta et al. [25] and showed that the expected coverage provided by their model is more suitable.

3.2. Single dispatch, total backup and non-homogeneous servers

Systems with non-homogeneous servers can be found in many real-world cases. For example, in an EMS, some ambulances only provide basic support; however, some of them can provide advanced support. Thus, when these two types of ambulances share a workload, the service rate will be the average service rate of basic and advanced ambulances. On the other hand, two systems with the similar servers in terms of vehicle, personnel and equipment may have different service rates based on their locations. For instance, the mean service time is changed due to the travel time, which is a function of servers' locations. Halpern [36] investigated the effects of depending the service time and customer location and dispatch units, in a simple two servers, two customers system.

Larson and McKnew [37] proposed the HQM and AH model for police patrol, where officers can be in one of three states, free, busy on PIA and busy assigned to a CFS, and therefore the total number of states in this study is $3^N$. They used the Larson’s approximation procedure to estimate the performance measures. In continuation of the previous research, McKnew [38] used a modified center of mass (MCM) dispatching policy in a police department with $3^N$ states. In this policy, the total service area is divided into several sub-
areas, and police cars are located at the center of mass of them. Each sub-area is distributed into atoms, and customers are located at the center of them. Upon arrival of a customer, the closest available car is assigned from the respective sub-area even if there is a closer car from other sub-areas. If all cars in a sub-area are busy, then the closest car from other sub-areas is assigned.

A common assumption in the hypercube model is to use a fixed-preference list to dispatch servers. This means that upon receiving a call, the first available server is assigned according to this list. Thus, if there are servers that have the same priority for a specific customer, a "tie" occurs. In AH models, it is assumed that there is only one server which is preferred to dispatch, but when multiple servers are located in one station, the tie occurs in the dispatch preferences with high probability. Modeling of such a situation by taking an arbitrary fixed-preference order leads to an imbalance in the workload among tied servers, because tied servers which are placed in the more preferred positions receive greater workload than those tied servers that are placed in the less preferred positions. Burwell [39] and Burwell et al. [40] proposed an “Internal Stacking” method to handle this situation. In their study, \( v_{jm} \) indicates the number of servers tied with server \( j \), including server \( j \), for the \( k^{th} \) preference position of customer \( m \). It is also assumed that these servers are positioned from \( k \) to \( (k - 1 + v_{jm}) \) in the dispatch list of customer \( m \). The set of these servers is given by \( H_{jm} \):

\[
H_{jm} = \left( \bigcup_{j=k}^{k+v_{jm}-1} \{ a_{ml} \} \right)^{-1}
\]  

(17)

Each time, if the \((k - 1)\) first servers are busy, then a server is selected randomly from \( H_{jm} \). The workload of server \( j \) must be calculated by conditioning on the number of tied servers, which are selected before server \( j \).

Brandeau and Larson [12] studied the effects of variable service times in the Larson's AH model. They also considered the mean service time calibration feature in this model and offered an algorithm to estimate the travel time effectively.

Takeda et al. [21] investigated ambulance decentralization in a case study and indicated that decentralization of ambulances can have positive impact on system performance measures. Budge et al. [13] proposed an approximation algorithm based on the Jarvis’ algorithm, in which more than one server can be assigned to each station; therefore, they computed a station (instead of server) busy probability. Also, in this algorithm, a set of correction factors is formulated based on random sampling of stations. They assumed that \( N \) servers are distributed among \( j \) stations, with \( n_j \) servers at station \( j \), and for all \( j \), \( n_j \geq 1 \). They also defined \( P_0 \) and \( P_N \) corresponded to the probability of the system being idle (all servers are available) and the probability of all servers being busy, respectively. This algorithm is started by calculating:

\( b_{k} \) = the \( k \)-th preferred station for node \( i \).
The number of servers at the $k$-th preferred station for node $i$.

$$n_{ki} = n_{bi} = \text{the number of servers at the } k\text{-th preferred station for node } i.$$  

$$z_{ki} = n_{ki} + n_{k2} + \ldots + n_{ki}$$

$$\tau_{ki} = \tau_{n_{ki}}$$

In this algorithm, $r_j$ corresponds to the busy fraction of each open station $j$, $\rho$ is the expected workload per server, and $r = \rho(1 - P_N)$ is the expected server utilization.

$$r_j = \frac{1}{n_j} \sum_{i=1}^{J} \lambda_i f_{ij} \tau_{ij}$$  \hspace{1cm} (18)

The station-specific correction factors can be expressed by:

$$Q\left(\{n_{wi}\}, \rho, k\right) = \frac{P^0_j \sum_{l=0}^{N-1} (\rho N)^{-l} \left( \prod_{u=0}^{z_{ki} - 1} \frac{s - u}{N - u} \right)}{r^{(k-1)} \left(1 - r_{ji}^0\right)}$$  \hspace{1cm} (19)

where $f_{ij}$ is the multi-server counterpart of Equation (8) and calculated by:

$$f_{ij} \approx Q\left(\{n_{wi}\}, \rho, k\right) \prod_{l=1}^{k-1} \left(1 - r_{ji}^0\right)$$  \hspace{1cm} (20)

In Equation (20), distribution of servers between stations affects the correction factor because the probability that a server from station $j$ is dispatched to a customer from node $i$ not only depends on the number of stations that are preferred (by node $i$) to station $j$, but also on the number of servers at those stations.

Next, by assuming that the system operates as an $M/M/N/N$ queuing system, the busy fractions and the system wide average service time are calculated by (superscripts are used as iteration counters):

$$\tau^0 = \frac{1}{\lambda N} \sum_{j=1}^{J} n_j \sum_{i=1}^{I} \lambda_i \tau_{ij}$$  \hspace{1cm} (21)

$$r_j^0 = r^0 = \lambda \tau^0 \frac{1 - P_N^0}{N}$$  \hspace{1cm} (22)

where $P_N^0$ is calculated using Erlang’s loss formula. Set the iteration counter, $h$, to one and repeat the following steps:

**Step 1)** Use $\tau^{h-1}$, $\lambda$ and $N$ to calculate $P_N^h$ and $P_N^h$.

**Step 2)** Calculate $V_j^h$ for all $j$ by using Equations (20) and (23):

$$V_j^h = \sum_{i=1}^{I} \lambda_i \tau_{ij} Q\left(\{n_{wi}\}, \rho^{h-1}, \alpha_i\right) \prod_{l=1}^{\alpha_i - 1} \left(1 - r_{ji}^{h-1}\right)$$  \hspace{1cm} (23)
where \( r^{h-1}_i \) or \( h \) is always the most recently computed station utilization (i.e., \( r^h_i \) ) if it has been computed, and otherwise \( r^{h-1}_i \). Then, update the station-specific busy fractions using Equation (24) if \( r^{h-1} \leq 0.5 \) and using Equation (25), otherwise:

\[
 r^h_j = \frac{V^h_j}{n_j + (r^{h-1}_j)^{n_j-1} V^h_j} \tag{24}
\]

\[
 r^h_j = \left( \frac{V^h_j}{(V^h_j/n_j + 1)^{n_j-1}} \right)^{1/n_j} \tag{25}
\]

**Step 3)** Calculate \( f^h_{ij} \) and normalize these probabilities using \( f^h_{ij} \leftarrow f^h_{ij} (1 - P^h_N) / \sum_{j=1}^J f^h_{ij} \) and then calculate \( \tau^h \), \( \rho^h \) and \( r^h \).

\[
 f^h_{ij} \approx Q\left( \{n_{ij}\}, \rho, k, \prod_{j=1}^{L} r^h_i \left( 1 - r^h_{ij} \right) \right) \tag{26}
\]

\[
 \tau^h = \frac{1}{\lambda (1 - P^h_N)} \sum_{j=1}^J \sum_{i=1}^I f^h_{ij} \tau_{ij} \tag{27}
\]

\[
 \rho^h = \frac{\lambda \tau^h}{N} \tag{28}
\]

\[
 r^h = \frac{1}{N} \sum_{j=1}^J n_j r^h_j \tag{29}
\]

**Step 4)** If \( |r^h_j - r^{h-1}_j| < \varepsilon \) for all \( j \), stop. Otherwise, set \( h = h+1 \).

Budge et al. [41] used this algorithm to find the relationship between travel time and distance. They concluded that a logarithmic transformation makes symmetric travel-time distribution. Additionally, Toro-Díaz et al [42] used this algorithm to extend the work given by Toro-Díaz et al [32] and presented a multi-objective location model to balance efficiency and fairness, where more than one server can be assigned to each station. In their work, the purpose of fairness is to make same mean response times and also same server’s workload. Ansari et al [43] used this approximation algorithm to estimate the correction factors, the average server workload and the individual server workload and treated them as constants in an MILP model. This model maximizes the number of high-priority calls that can be covered within a time threshold (i.e., the coverage level) and balanced server workload by determining the location of ambulances and dispatching policy simultaneously. They proposed an iterative algorithm to solve a real-world example and concluded that the server workload maintain equivalence by a small reduction in coverage.

Boyaci and Geroliminis [44] proposed two extensions for the AH model, in which more than one server can be assigned to each atom. In the first extended AH model, it is assumed that the service rate is equal for the intra and inter-district customers; therefore, each server
has two states, free and busy. Also, in this model, each number in a system state corresponds to the number of busy servers in the corresponding atom. For example, state \( \{302\} \) stands for a state where 3, 0 and 2 servers are busy in the first, second and third atoms, respectively. Thus, if \( n_i \) shows the number of servers assigned to atom \( i \), and \( M \) indicates the total number of atoms, then there are \((n_1 + 1)(n_2 + 1)\cdots(n_M + 1)\) states. As usual, the steady-state probabilities in this system are computed by flow-balance equations. As an instance, transition equation for state \( \{302\} \) is written as (30) and Fig. (5) shows this transition network.

\[
P\{302\}(\lambda_1 + \lambda_2 + \lambda_3 + 3\mu_1 + 2\mu_2) = \lambda_3 P\{301\} + \lambda_1 P\{202\} + 3\mu_3 P\{303\} + \mu_2 P\{312\} + 4\mu_1 P\{402\}
\]

(30)

In the second extended AH model, the service rate is different for inter \( \mu_j \) and intra \( \mu'_j \) arrivals, and thus, each server has three states, free, busy serving an intra-district customer and busy serving an inter-district customer. A new definition for the states of the system is proposed, in which the state of each server is shown by two numbers. For example, in a system with two servers in each atom, state \( \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \) indicates a situation, in which two servers in the first atom are busy (according to the first row), and one of them serves an intra-district customer and the other one serves an inter-district customer. Similarly, one server of the second atom is busy and serves an intra-district customer, and the other one is free. Therefore, this system has \( \prod_j \left( n_j + \frac{2}{2} \right) \) states. The transition rate of state \( \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \) is as Equation (31) and Fig. (6) shows this transition network.

\[
P_{\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}}(\lambda_1 + \lambda_2 + \mu_1 + \mu'_1 + \mu_2) = \lambda_1 P_{\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}} + \lambda_2 P_{\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}} + \mu_2 P_{\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}} + 2\mu_1 P_{\begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}}
\]

(31)

Finally, they showed that Monte Carlo sampling is applicable for the HQM and can represent its features and solve its steady-state probabilities. Boyaci and Geroliminis [45] proposed a mixed-hypercube queuing algorithm (MHQA) in an emergency system, which can generally be presented in three steps. In the first step, the total service area is divided into the sub-areas iteratively until the size of each sub-problem becomes solvable. In the second step, these sub-problems are solved by considering three states for each server (i.e., the second extended AH model in Boyaci and Geroliminis [44]). In the final step, sub-areas are merged by using an
approximate hypercube model, in which some servers have two states and the others have three states, because servers located near the borders between two merged areas can provide service to both areas at the same rate. This algorithm computes the lost rate for the main service area. Boyaci and Geroliminis [46] proposed a partitioning algorithm to obtain more accurate results in the MHQA.

Baptista and Oliveira [47] extended the AH model, in which the customer arrival process is not stationary. However, service time, response time and preference list are independent of time periods. It is obvious that the AH model can be applied only at periods, in which the arrival rate is stationary; therefore, the average unit workloads is computed by:

$$\rho_q = \frac{\sum_{q \in J} t_q \rho_j^q}{\sum_{q \in J} t_q}$$  \hspace{1cm} (32)

where,

- $T$ number of time periods
- $r_t$ time period $t \ (t = 1, ..., T)$
- $q$ set of periods $r_t$ where the arrival rate is stationary, i.e. $q = r_1, r_2, ..., r_q$ and $q \in J$
- $j$ the set of the $q$ different stationary periods
- $t_q$ number of time periods $r_t$ in set $q$
- $\rho_j^q$ workload of server $j$ in period $q$

They estimated a set of the system performance measures using the presented hypercube model and used a simulation model in a case study to assess the validity of this AH model based on four dispatching rules: 1) nearest neighbor rule: servers are ranked based on their distances to the customers, 2) less occupied preference rule: servers are ranked based on the number of times that each server is assigned, 3) area preferred server rule: only a server with the highest priority can be assigned to serve a customer, and if it is not available, a server is assigned from another ESS, and 4) area two most preferred servers: only two servers with the highest priority can be assigned to respond to a customer in order, and if neither of them is available, a server is assigned from another ESS.

Iannoni et al. [48] suggested an HQM to analyze the EMS, in which customers have different priorities. In this study, low-level customers are kept waiting until the number of idle servers reaches the threshold number (i.e., cut-off level) to increase the probability of serving the higher-level customers immediately, upon arrival. They calculated the performance measures for this cut-off HQM.

As shown in Section 2-1, the studies proposed location models in a hypercube framework are as follows. Goldberg and Paz [49] modified the model presented in Goldberg et al. [30] by considering the FIFO queue discipline, customer classification and allocation of multiple servers at each base location. They tested the applicability of the model in more real test problems and proposed a heuristic algorithm. Zhu and McKnew [50] proposed a workload balancing allocation model (WBAM) to deploy a number of ambulances and balance workload between servers. This model uses a goal programming approach to address this aim
and calculates server workload with an AH model developed by Burwell [39]. Lei et al [51] formulated a four-objective model for a districting-routing problem under dynamic and stochastic conditions. They solved this model by a two-stage stochastic programming approach and an enhanced multi-objective evolutionary algorithm.

Geroliminis et al. [52] proposed a hybrid queuing location model to minimize the mean response time and meet the minimum coverage level. Actually, they extended the MCLP for locating servers and used the HQM for districting and dispatching purposes. In this study, server response time depends on the customer's location, and service rate for intra-district customers located in the server's region is lower than that for customers out of that region (i.e., inter-district). Later, Geroliminis et al. [53] extended the previous model for locating emergency vehicles on urban networks with many servers, subject to hypercube flow-balancing equations. This model will be described in Section 3. Geroliminis et al. [54] used a GA combined with the hypercube model to solve this model in a two-stage approach. In the first stage, the overall service area is districted into subareas, and a number of servers are allocated to each subarea. In the second stage, the optimal location of servers is determined in their subarea. In both stages, the AH model is used to evaluate the fitness function. The results of the model application indicated that this model is suitable as an optimization tool, particularly when many servers must be located.

Erkut et al. [55] extended ten existing covering models for emergency systems by taking the survival function, which maximizes the expected number of patients who survived cardiac arrest. They used the Jarvis' algorithm to evaluate this function. Ingolfsson et al. [56] designed a location model to minimize the number of ambulances and satisfy the minimum threshold of the service level. In this paper, the service level is determined by the number of customers responded to in a time interval. They also considered random pre-trip delays in addition to random travel times. They used the approximation procedure proposed by Budge et al. [13] to evaluate the server's busy fraction.

McLay [57] proposed an MEXCLP with two types of servers and multiple types of customers (MEXCLP2) to determine the locations of ambulances optimally. The goal of their model is to maximize the expected number of customers in life-threatening situations covered within a specified time. To calculate server busy probabilities, they extended an approximation algorithm based on the Jarvis’ algorithm for a case with an infinite queue.

Rajagopalan and Saydam [58] formulated minimum expected response location problem (MERLP) to determine the locations of ambulances in order to minimize the expected response distances and meet minimum coverage requirements. They incorporated the concept of coverage in their model by using the Daskin's expected coverage [24] as presented in Section 2-1 and the Marianov and Revelle’s available coverage [59], in which only those customers are incorporated in the coverage statistics whom covered with a pre-determined reliability. They also incorporated server busy probabilities, which are computed by the Jarvis' algorithm, in the expected and available coverage statistics. By applying the MERLP to a case study, they compared this model with the MEXCLP [24] and showed that in MERLP, the response time is faster and therefore more lives can be saved.
3.3. Single dispatch, partial backup and non-homogeneous servers

When assuming that there is partial cooperation between servers, some servers cannot respond to some customers for reasons, such as the location of the customer or the type of the customer's demand. In these systems, on arrival of a customer, if the backup servers are busy, the customer is lost or enters a queue even if there are other free servers.

To better understand the presented concepts, the reader may consider the example provided in Section 1. Assume that there is no way between atoms 2 and 3 (Fig. 7). Therefore, the preference matrix based on the shortest travel distance is changed as shown in Table 5.

\[ \text{(Please insert Fig. 7 about here.)} \]
\[ \text{(Please insert Table 5 about here.)} \]

Now, the balance equation for state \{011\} is as follows. The system leaves state \{011\} if a customer arrives from atom 1 or 2 or server 2 or 3 completes its service, so the transition rate is \((\lambda_1 + \lambda_2 + \mu_2 + \mu_3)P\{011\}\). Moreover, the system enters this state in one of the following three ways (in accordance with Table 4): i) from state \{001\} if a customer arrives from atom 2 (it is noteworthy that in this state, if a customer arrives from atom 3, it has been lost or served by another system, although servers 1 and 2 are idle), ii) from state \{010\} if a customer arrives from atom 3, and iii) from state \{111\} when the service of server 1 is completed. The transition rate is

\[(\lambda_2)P\{001\} + \lambda_3 P\{010\} + \mu_4 P\{111\}\].

Since the balance equation of state \{011\} is written by:

\[ (\lambda_1 + \lambda_2 + \mu_2 + \mu_3)P\{011\} = (\lambda_2)P\{001\} + \lambda_3 P\{010\} + \mu_4 P\{111\} \] (35)

Mendonca and Morabito [60] investigated ambulance deployment on a highway, which connects the cities of Sao Paulo and Rio de Janeiro in Brazil. In this study, only a part of the highway is analyzed, which is covered by the Anjos do Asfalto’s emergency system. This EMS has six ambulance bases (i.e., \(2^6\) states) along the highway, and one ambulance is stationed at each base. The central dispatcher is located in Rio de Janeiro and records all the movements of ambulances, even for fuelling. When an emergency call is received, the nearest ambulance is dispatched to the place of incident, and if it is busy, the second nearest ambulance is dispatched. If this ambulance is busy too, then the customer is lost and transferred to another EMS. They used a hypercube model to evaluate the performance measures of this system, such as mean response time and workload of ambulances. They showed that only by changing the sizes of the atoms, the ambulance workload becomes more balanced.

Atkinson et al. [61] proposed one exact and two heuristic methods to estimate loss probabilities and ambulance utilization rate for the EMS studied by Mendonca et al. [62]. They also showed the accuracy of the proposed heuristic methods with numerical example. Atkinson et al. [63] extended these two heuristic methods for a system with \(3^n\) states: free, busy serving a first-preference customer and busy serving a second-preference customer.
Iannoni et al. [64] presented several greedy heuristic algorithms to optimize ambulance location and dispatch policies for the EMS presented in Mendonca et al. [62]. They embedded in each optimization procedure the hypercube approximation algorithm proposed by Atkinson et al. [62], to solve large-scale problems fast and with acceptable precision.

Morabito et al. [65] investigated the effects of taking into account homogeneous against non-homogeneous servers in calculating the performance measures with HQM. They concluded that even when the level of non-homogeneity is not significant, it leads to different results to consider it in performance measures. Kim and Lee [66] used the HQM to compute steady-state probabilities in a probabilistic location set covering problem (PLSCP) to satisfy the reliability requirements. In the PLSCP presented by Revelle and Hogan [67], the total number of ambulances is minimized while the location of ambulances are determined in such a way that the number of ambulances that cover each node will be higher than minimum requirement. They also suggested two iterative optimization algorithm based on the HQM and simulation and found that the performance of these two algorithms is almost equivalent.

3.4. Multiple dispatch

As noted above, in the original model, only one server is dispatched to serve a customer; however, in real-world situations (e.g., large fire or severe accidents), more than one server is usually dispatched. This section presents the studies with multiple dispatch assumption in hypercube models, in which some customers require the simultaneous dispatching of two or more servers.

Now assume that in the example presented in Section 1, customers from each atom $j$ can be of two types. Type 1 customers require service by a server with arrival rate $\lambda_j^{[1]}$, and type 2 customers require the simultaneous service by two servers with arrival rate $\lambda_j^{[2]}$. The total arrival rate of the system is as follows:

$$\lambda = \lambda_j^{[1]} + \lambda_j^{[2]} + \lambda_j^{[1]} + \lambda_j^{[2]} + \lambda_j^{[1]} + \lambda_j^{[2]}$$

(S36)

Suppose that each server can be dispatched to each atom (total backup) and when a type 1 customer asks for service, a server with the highest priority among the available servers (Table 2) will be dispatched. In the case of the type 2 customer, the first two preferred servers are dispatched simultaneously. If one of them is busy, the third server from priority list is sent. When only one of the three servers is available, it is assigned as a single dispatch. As a result, in addition to the transitions which occur on the cube edges (Fig. 1), some upward transitions can occur on the diagonals of this cube, such as: $\{000\} \rightarrow \{110\}$, $\{100\} \rightarrow \{111\}$, $\{010\} \rightarrow \{111\}$.

The mean service rate for the type 1 customer served by server $i$ is $\mu_i$. In the case of the type 2 customer, the model considers two servers that are servicing simultaneously the same customer, as same as two independent servers service two separated type 1 customers. Thus, when a type 2 customer is served by servers $i$ and $k$, the mean service rates are $\mu_i$ and $\mu_k$, respectively.
Now, the balance equation for state \{110\} is as follows. The system leaves this state if a customer arrives or server 1 or 2 completes its service, so the transition rate is \((\lambda + \mu_1 + \mu_2)P\{110\}\). Moreover, the system enters this state in one of the following four ways (in accordance with Table 4): i) from state \{000\} if a type 2 customer arrives from atom 2, ii) from state \{010\} if a type 1 customer arrives from atom 1 or 2, iii) from state \{100\} when a type 1 customer arrives from atom 2, and iv) from state \{111\} when a service of server 3 is completed. Since the balance equation of state \{110\} is written by:

\[
(\lambda + \mu_1 + \mu_2)P\{110\} = \lambda_2^{[2]}P\{000\} + (\lambda_1^{[1]} + \lambda_2^{[1]})P\{010\} + \lambda_2^{[2]}P\{100\} + \mu_3P\{111\}
\]

Chelst and Barlach [68] studied an example of multiple dispatch HQM in a police patrol system for the first time. They proposed the HQM and AH models for ESSs, in which two servers can be dispatched together to one customer; however, these servers are homogeneous. Then, they estimated the performance measures of this system by the exact and approximate models. In the approximate model, they assume that:

- \(U_j\) = the \(j\)-th ranked server is unavailable
- \(F_j\) = the \(j\)-th ranked server is available (free)

\[
P(U_j) = P(U_2 \ldots U_{j-1} F_j) = \text{probability that the } j\text{-th ranked server will be dispatched to a type 1 customer, which requires service by a server.}
\]

\[
P(U_1 \ldots U_{m-1} F_m U_{m+1} \ldots U_{j-1} F_j) = \text{probability that the } j\text{-th and } m\text{-th ranked servers simultaneously dispatched to a type 2 customer, which requires the simultaneous service by two server.}
\]

For a set of identical servers, we have:

\[
P(U_j) = Q(N, j-1, 1) \rho^{j-1} (1 - \rho) \quad (38)
\]

\[
P(U_j) = Q(N, j-1, 2) \rho^{j-2} (1 - \rho)^2 \quad (39)
\]

where \(Q(N, j-1, 1)\) and \(Q(N, j-1, 2)\) are the correction factor for type 1 and 2 customers. The first correction factor is calculated like Larson’s correction factor (Eq. 4); however, unlike Larson’s models, there is not a simple closed form for \(P(S_k)\) (the reader may refer to Chelst and Barlach [68], Appendix II),

\[
Q(N, j-1, 1) = \sum_{k=j-1}^{N} \binom{k}{j-1} \binom{N-k}{N-(j-1)} \frac{P(S_k)}{\rho^{j-1} (1 - \rho)} \quad (40)
\]
To find the second correction factor, suppose that \( k \) out of \( N \) servers are busy. The probability that only one server out of the first \( (j-1) \) servers is free is as follows:

\[
\sum_{k=2}^{N-1} \binom{k}{j-2} \binom{N-k}{1} \binom{N}{j-1} \tag{41}
\]

The probability that the free server is the \( m \)-th ranked server is \( 1/(j-1) \) and is not dependent on \( m \). Therefore, the probability that only the \( m \)-th server out of the first \( (j-1) \) servers is free is as follows:

\[
\sum_{k=2}^{N-1} \binom{k}{j-2} \binom{N-k}{1} \binom{N}{j-1} (j-1) \tag{42}
\]

The conditional probability that the \( j \)-th server is available is as follows:

\[
(N-k-1)/N-(j-1) \tag{43}
\]

And Equation (39) becomes as follows:

\[
P(U_1 U_2 \ldots U_{k-1} F_k U_{k+1} \ldots U_{j-1} F_j) = \sum_{k=2}^{N-1} \binom{k}{j-2} \binom{N-k}{1} \binom{N}{j-1} \binom{N-k-1}{j-1} P(S_k) \tag{44}
\]

The second correction factor is calculated by Equations (40) and (44) and is independent of \( m \).

\[
Q(N, j-1, 2) = \sum_{k=2}^{N-1} \binom{k}{j-2} \binom{N-k}{1} \binom{N-k-1}{j-1} P(S_k) \tag{45}
\]

Davoudpour et al. [69] introduced a probabilistic coverage model that integrates the MEXCLP and hypercube queuing model. They indicated the applicability of this model by applying it onto an EMS center in Tehran with two basic support and two advanced support ambulances. Because of the small size of their problem, they solved steady-state equations to calculate state probabilities. They concluded that the number of servers in the center has a large effect on the number of customer, which has been responded. Then, they showed that the relationship between this performance measure (i.e., system responsiveness) and the parameters of the system is linear. Sudtachat et al [70] tried to maximize the patient survival
probability in a system with two types of ambulances, basic life support (BLS) and advanced life support (ALS). They also considered three priorities for customers based on their severity level, which is determined at first by the dispatcher and can be updated when the server arrives on-scene. The priority 1 customers need simultaneously to serve by two servers, an ALS and a BLS and the priority 2 customers are served by the closest available BLS. The priority 3 customers need one BLS which is selected according to an ordered preference list because this type of customers are considered non-critical and sending the closest BLS unit to them may make ambulances unavailable for next life-threatening customers. They developed a simulation model for small problems and proposed a heuristic algorithm based on the AH model for large-scale problems to design dispatching strategies. They concluded that dispatching based on customer priorities improves patient survival probability rather than dispatching based on the closest strategy. They also showed that the number of ALS units and their location are important factors on the efficiency of the heuristic policy.

Iannoni and Morabito [71] formulated an HQM that simultaneously takes into account various assumptions including different types of customers and servers, partial backup, single or multiple dispatch and a third state for servers. They stated that these extended HQM can be embedded directly into an optimization procedure and are suitable for evaluating the performance measures. Iannoni et al. [72] embedded HQM into a GA algorithm to optimize the size of each atom for the model presented in Iannoni and Morabito [71]. In this GA/hypercube algorithm, each generated configuration (represented by a chromosome) is evaluated by the hypercube model. They verified that this GA/hypercube algorithm is effective to calculate the performance measures and showed that these measures can be improved only by modifying the atom sizes of the system and it is not necessary to relocate the ambulances and additional investments on capacity. Iannoni et al. [73] used this algorithm to determine the locations of ambulances and their coverage areas to minimize the response time and imbalances in the ambulance workload. In this study, ambulance bases can be located anywhere along the highways. Table 6 presents a List of studies in the literature with focus on their assumptions.

4. Location models and solution approaches

In recent years, the increasing costs of emergency service, high volume of emergency calls and traffic problem have made the location-allocation problem as a major issue in designing emergency service systems. On the other hand, the server performance is related to the dispatching and districting policy, which are dependent upon the locations of emergency facilities and allocation of servers. Also, there are usually a limited number of servers that must be allocated to the facilities to ensure the adequate coverage and appropriate response time. Most studies in the literature tried to integrate location models with HQM because this model is able to assess the potential node of facility location from different perspectives. Table 7 represents a summary of these studies. As shown in this table, most studies used approximate hypercube queuing model since this model is solved easier than the exact model.
Batta et al. [25] showed that using the AHQM in location models usually overestimates the coverage and underestimates the number of required servers.

{Please insert Table 7 about here.}

Geroliminis et al. [52-54] and Toro-Diaz et al. [32] developed a model in the framework of hypercube to optimize the performance measures in large urban networks with many servers, without using approximate approaches. The objective of this model is to minimize the mean system response time, subject to hypercube constraints. The general structure of these models is as follows:

\[
\min z = \sum_{j=1}^{N} \sum_{m=1}^{M} \rho_{jm} I_{jm} 
\]

\[\text{s.t.} \]
\[
\sum_{i \in W_m} x_i \geq y_m, m = 1, 2, \ldots, M 
\]
\[
\sum_{i=1}^{I} x_i = N 
\]
\[
x_i \in \{0, 1\}, i = 1, 2, \ldots, I 
\]
\[
y_m \in \{0, 1\}, m = 1, 2, \ldots, M 
\]
\[
\rho_{jm} = h_m \frac{\sum_{V_i \in E_m} P\{V_i\}}{1 - P\{V_{2^N-1}\}}, j = 1, \ldots, N; m = 1, \ldots, M 
\]
\[
P(V_m) \left[ \sum_{V_i \in C_{m1}} \mu_{im} + \sum_{V_i \in C_{m2}} \lambda_{im} \right] = \sum_{V_i \in C_{m1}} \mu_{im} P\{V_i\} + \sum_{V_i \in C_{m2}} \lambda_{im} P\{V_i\}, m = 1, \ldots, 2^N - 1 
\]
\[
\sum_{i=0}^{2^N-1} P\{V_i\} = 1 
\]

there are two decision variables:

\[
x_i = \begin{cases} 
1 & \text{if a facility is located at potential site } i \\
0 & \text{otherwise} 
\end{cases} 
\]
\[
y_m = \begin{cases} 
1 & \text{if demand point } m \text{ is covered} \\
0 & \text{otherwise} 
\end{cases} 
\]

where \( P\{V_k\} \) is the steady-state probability of the state corresponding to vertex \( V_k, k = 0, 1, \ldots, 2^N - 1 \); \( d_m^- \) and \( d_m^+ \) include the downward and upward Hamming distances.
between vertices \( V_i \) and \( V_m \): the Hamming distance between two vertices \( V_i \) and \( V_m \) is the number of digits that are different between two vertices. For example, the Hamming distance between states \( \{0011\} \) and \( \{0111\} \) is equal to one, but the distance between states \( \{0110\} \) and \( \{1001\} \) is equal to 4. \( d_{im}^- \) and \( d_{im}^+ \) include the number of digits changed from 0 to 1 and from 1 to 0, respectively. Given that the system is in state \( i \), \( \lambda_{ij} \) and \( \mu_{ij} \) are the upward and downward rates to transition from state \( i \) to state \( j \) corresponding to vertices \( V_i \) and \( V_m \).

In this model, Constraint (46) specifies the demand points to be covered. Constraint (47) specifies the number of facilities that must be deployed. Constraints (48) and (49) are integrality constraints for the decision variables. Constraint (51) calculates \( \rho_{jm} \), where its denominator shows the probability that all servers are not busy. Constraint (51) shows the set of flow-balancing equations, which compute the probability of states. Actually, these equations are the modified versions of the original HQM [2] to take into account different service rates for inter-district and intra-district customers. Constraint (52) guarantees that the sum of probabilities is equal to one. This model is in fact a two-step model, in which the service area is distributed into sub-areas in the first step, and, in parallel, the required number of servers is assigned to each sub-area. In the second step, the optimal locations of servers in their atoms are determined.

5. Conclusion and future research

This study reviewed the literature on hypercube queuing model with focus on the research published after Larson [2]. These studies were classified with respect to their assumptions, such as dispatch policy, backup policy and the homogeneity of servers. The growing attention into the hypercube model in comparison with simulation approaches are due to its application in real-world problems. Actually, the HQM can reflect various aspect of an emergency service system and describe the states of these systems as well. However, the existing models are far from real-world situations and still much work remains to be done. As an example, Souza et al. [19] discussed the provision of a model to subdivide a service area into smaller sub-areas optimally. Davoudpour et al. [69] considered a number of uncertain parameters, such as demand rate and service time. Chiyoshi et al. [3] suggested that different demand rates should be considered during a day or a week. They also offered multiple dispatching, in which servers are not sent simultaneously.

The authors of this paper suggest that eliminating the server's returning time can improve the HQM. The main purpose of an emergency system is to provide service as fast as possible. In real-world examples, the response time is reduced by eliminating the server's returning time to its base. Also, when a customer enters during the time of returning, the server can go to the customer's location directly. Because of the complexity of modeling such assumptions, the existing studies have supposed that a server returns to its base when a service is completed, and will then be dispatched to serve a queued customer. Therefore, taking into account these assumptions helps to calculate the performance measures more accurately. On the other hand, in EMS papers, there are no various policies for returning servers and most
articles suppose that servers should come back to their home base station, but this is not necessary. Sometimes servers should come back to another station for better coverage. It seems that new state definition in the HQM models should have done for this repositioning problem to save the lives of more people.

Furthermore, due to the complex nature of the problems in this field, many studies have used a fixed procedure to dispatch servers. In real-world problems, server dispatch and backup policies are dependent upon the conditions of the system, such as the real-time location of the server, demand fluctuation and the type of the customer's demand. Therefore, it is recommended that a flexible procedure be defined to reflect these conditions. In view of the aforementioned literature and Table 7, which summarizes the studies that have tried to integrate the HQM and location model, a number of suggestions for future studies are given below.

- As noted before, to design an ESS, decisions are divided into two types: strategic decisions to determine the number and locations of servers and tactical decisions to specify dispatching policies and server coverage areas. As it turns out, strategic and tactical decisions are related to each other. Thus, the location model and HQM should be integrated to increase their effectiveness. The studies presented so far have not been successful in fully integrating the two models and are, in the best case, designed in two phases, and there is no relationship between the state probabilities and decision variables of the location model. In fact, the balance equations have not been intended in location models explicitly, and existing models use an approximate hypercube or are designed in two phases. Therefore, it is a promising area for the further study to design a model that can provide a relationship between balance equations and location variables.

- As it can be seen in Table 7, the objective functions of most location models are either travel time minimization or coverage maximization. Furthermore, all of these studies have only a single objective function. Despite the humanitarian nature of these problems, the economic concern is noteworthy, because in these types of problems, available financial resources are very restricted. Thus, models with multiple objectives considering economic and performance measures are more effective. Additionally, environmental measures can be regarded.

- Due to the significance of travel time as a portion of service time, many studies have tried to analyze this time more accurately [12, 18, 30, 41]; however, travel time is always evaluated as a part of the service time. Therefore, it can be very helpful to define a measure to examine the travel time solely. It seems reasonable to consider separate rates for travel time and on-scene time, because travel time is dependent on factors (e.g., distance and type of vehicles) while on-scene time depends on the expertise of the medical team and severity of the incident. It is recommended that distinct distributions with different rates for travel time and on-scene time can be used.

References


Information about the Authors
Ms. Maryam Ghobadi is a Ph.D. Candidate from 2014 in Kurdistan University in Iran at the faculty of Engineering. She got her M.Sc. degree in Industrial Engineering from Alzahra University in 2013. In research, her current interests include Location Problem and Queuing System. Ms Ghobadi has published a book in the context of Location Problem.

Dr. Jamal Arkat received BSc, MSc, and PhD degrees in Industrial Engineering from Iran University of Science And Technology (IUST) in Iran. He is currently an Associate Professor in Department of Industrial Engineering at University of Kurdistan, Sanandaj, Iran. His research interests include operations research applications in health care, crisis management and stochastic processes. He is now the vice dean for research in Faculty of Engineering at University of Kurdistan.

Dr. Reza Tavakkoli-Moghaddam is a professor of Industrial Engineering at University of Tehran in Iran. He obtained his Ph.D. in Industrial Engineering from the Swinburne University of Technology in Melbourne (1998). He is an Associate Member at Academy of Sciences in Iran and serves as Editorial Board of the International Journal of Engineering and Iranian Journal of Operations Research. He was the recipient of the 2009 and 2011 Distinguished Researcher Award and the 2010 Distinguished Applied Research Award at the University of Tehran, Iran. He was selected as National Iranian Distinguished Researcher for 2008 and 2010 in Iran. Professor Tavakkoli-Moghaddam has published four books, 15 book chapters, more than 500 papers in reputable academic journals and conferences.
Figure Captions

Fig. 1. State space of a system with three servers

Fig. 2. Network with three atoms [3]

Fig. 3. Divisions of service time [12]

Fig. 4. Transition into and out of state \{ab\} [19]

Fig. 5. Transition network for state \{302\} with equal inter and intra-district service rates

Fig. 6. Transition network for state \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} with different inter and intradistrict service rates

Fig. 7. Network with three atoms
List of Figures

Fig. 6. State space of a system with three servers.

Fig. 7. Network with three atoms [3].

Fig. 8. Divisions of service time [12].
Fig. 9. Transition into and out of state \( \{ab\} \) [19].

Fig. 10. Transition network for state \( \{302\} \) with equal inter and intra-district service rates.

Fig. 6. Transition network for state \( \left\{ \begin{array}{c} 1 \\ 1 \\ 0 \end{array} \right\} \) with different inter and intra district service rates

Fig. 7. Network with three atoms
Table Captions

**Table 1.** Travel distance matrix between atoms

**Table 2.** Server dispatch preferences

**Table 3.** Server dispatch preferences

**Table 4.** Example of the dispatching matrix

**Table 5.** Server dispatch preferences

**Table 6.** List of studies in the literature with focus on their assumptions

**Table 7.** List of studies in the literature with focus on location models
List of Tables

Table 1. Travel distance matrix between atoms

<table>
<thead>
<tr>
<th>From/To</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2. Server dispatch preferences

<table>
<thead>
<tr>
<th>Atom</th>
<th>Server preference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Table 3. Notations used in this paper

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Number of servers ($j = 1, \ldots, N$)</td>
</tr>
<tr>
<td>$M$</td>
<td>Number of demand points ($m = 1, \ldots, M$)</td>
</tr>
<tr>
<td>$C$</td>
<td>Type of customer ($c = 1, \ldots, C$)</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Average system busy probability</td>
</tr>
<tr>
<td>$\rho_j$</td>
<td>Busy probability of server $j$</td>
</tr>
<tr>
<td>$\rho_{jm}$</td>
<td>Fraction of dispatches where server $j$ is sent to atom $m$</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>System-wide arrival rate</td>
</tr>
<tr>
<td>$\lambda_c$</td>
<td>Arrival rate of customers from node $c$</td>
</tr>
<tr>
<td>$\tau$</td>
<td>System-wide mean service time</td>
</tr>
<tr>
<td>$\tau_{jc}$</td>
<td>Expected service time for server $j$ and a customer of type $c$</td>
</tr>
<tr>
<td>$f_{jc}$</td>
<td>Probability that a customer of type $c$ is assigned to server $j$</td>
</tr>
<tr>
<td>$h_m$</td>
<td>Proportion of demand that is generated at node $m$</td>
</tr>
<tr>
<td>$t_{ij}$</td>
<td>Expected travel time between customer $m$ and server $j$</td>
</tr>
<tr>
<td>$W_m$</td>
<td>Set of potential sites covering demand point $m$</td>
</tr>
<tr>
<td>$S_k$</td>
<td>A state in which exactly $k$ servers are busy</td>
</tr>
<tr>
<td>$P(S_k \text{ or } P_N)$</td>
<td>Probability that the system is in state $S_k$</td>
</tr>
<tr>
<td>$P(S_0 \text{ or } P_0)$</td>
<td>Probability that all servers are idle</td>
</tr>
<tr>
<td>$P(S_N \text{ or } P_N)$</td>
<td>Probability that all servers are busy</td>
</tr>
<tr>
<td>$a_{ck}$</td>
<td>Index of the $k^{th}$ preferred server for customers of type $c$</td>
</tr>
<tr>
<td>$B_j$</td>
<td>The event that the $j^{th}$ selected server is busy</td>
</tr>
<tr>
<td>$F_j = B^c_j$</td>
<td>The event that the $j^{th}$ selected server is free</td>
</tr>
<tr>
<td>$P(V_k)$</td>
<td>Steady-state probability of the state corresponding to vertex $V_k$</td>
</tr>
<tr>
<td>$E_{jm}$</td>
<td>Set of states where server $j$ is the nearest available server to customer $m$</td>
</tr>
<tr>
<td>$C_N$</td>
<td>Vertices of $N$-dimensional unit hypercube</td>
</tr>
<tr>
<td>$d_{m}^{-}, d_{m}^{+}$</td>
<td>Downward and upward Hamming distances between vertices $V_i$ and $V_m$</td>
</tr>
<tr>
<td>$Q(N, \rho, j)$</td>
<td>Larson's correct factor</td>
</tr>
</tbody>
</table>

Table 4. Example of the dispatching matrix

<table>
<thead>
<tr>
<th>Atom</th>
<th>Layer</th>
<th>Dispatch preference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1&lt;sup&gt;st&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>b</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>2</td>
</tr>
</tbody>
</table>
Table 5. Server dispatch preferences

<table>
<thead>
<tr>
<th>Atom</th>
<th>1&lt;sup&gt;st&lt;/sup&gt;</th>
<th>2&lt;sup&gt;nd&lt;/sup&gt;</th>
<th>3&lt;sup&gt;rd&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Category</td>
<td>Type of customer</td>
<td>No queue</td>
<td>FIFO</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>Single dispatch, total backup, homogeneous servers</td>
<td>Identical customer</td>
<td>Larson [2], Larson and Franck [10], Chelst and Jarvis [17], Berman et al [22], Larson and Rich [18], Chiyoshi et al [26], Rajagopalan et al [33], Galvao et al [27], Chiyoshi et al [3], Toro-Diaz et al [32], Saydam et al [34], Sudtachat et al [35]</td>
<td>Larson [14], Batta et al [25]</td>
</tr>
<tr>
<td>Single dispatch, total backup, non-homogeneous servers</td>
<td>Identical customer</td>
<td>Brandeau and Larson [12], Zhu and McKnew [50], Geroliminis et al [52], Erkut et al [55], Rajagopalan and Saydam [58], Geroliminis et al [53], Budge et al [41], Geroliminis et al [54], Toro-Diaz et al [42]</td>
<td>Larson and McKnew [37]</td>
</tr>
<tr>
<td>Non-identical customer</td>
<td>Halpern [36], McKnew [38], Jarvis [15], Burwell [39], Burwell et al [40], Ingolfsson et al [52], Budge et al [13], Boyaci and Geroliminis [45], Boyaci and Geroliminis [46], Ansari et al [43]</td>
<td>Goldberg and Paz [49], Takeda et al [21], McLay [57], Baptista and Oliveira [47]</td>
<td>-</td>
</tr>
<tr>
<td>Single dispatch, partial backup, homogeneous servers</td>
<td>Identical customer</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Non-identical customer</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Single dispatch, partial backup, non-homogeneous servers</td>
<td>Identical customer</td>
<td>Kim and Lee [66]</td>
<td>-</td>
</tr>
<tr>
<td>Non-identical customer</td>
<td>Mendonca and Morabito [60], Atkinson et al. [61], Morabito et al [65]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>multiple dispatch, total backup, homogeneous servers</td>
<td>Identical customer</td>
<td>Chelst and Barlach [68]</td>
<td>-</td>
</tr>
<tr>
<td>Non-identical customer</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>multiple dispatch, total backup, non-homogeneous servers</td>
<td>Identical customer</td>
<td>Davoudpour et al [69], Sudtachat et al [70]</td>
<td>-</td>
</tr>
<tr>
<td>Non-identical customer</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>multiple dispatch, partial backup, homogeneous servers</td>
<td>Identical customer</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Non-identical customer</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>multiple dispatch, partial backup, non-homogeneous servers</td>
<td>Identical customer</td>
<td>Iannoni and Morabito[71], Iannoni et al [72], Iannoni et al [73]</td>
<td>-</td>
</tr>
<tr>
<td>Non-identical customer</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Reference</td>
<td>Hypercube</td>
<td>Exact</td>
<td>Approximate</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Berman et al [22]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Berman and Larson [23]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Battat al et [25]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Goldberg et al [30]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Goldberg and Paz [49]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Geroliminis et al [52]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Rajagopalan et al [33]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Galvao et al [27]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Erkut et al [55]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Ingolfisson et al [52]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>McLay [57]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Rajagopalan, and Saydam [58]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Geroliminis et al [53]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>McLay and Mayorga [31]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Geroliminis et al [54]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Toro-Diaz et al [32]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Saydam et al [34]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Davoudpour et al [69]</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Reference</td>
<td>Hypercube</td>
<td>Location model</td>
<td>Objective function</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>----------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>Toro-Diaz et al</td>
<td></td>
<td></td>
<td>Minimization of the mean response time and maximization of the fairness</td>
</tr>
<tr>
<td>[42]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ansari et al</td>
<td></td>
<td></td>
<td>Maximization of the coverage</td>
</tr>
<tr>
<td>[43]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kim and Lee</td>
<td>PLSCP</td>
<td></td>
<td>Minimization of the number of ambulances</td>
</tr>
<tr>
<td>[66]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>